

File created: 19-Jan-93 10:24:59 {DSK}<python>lde>lispcore>sources>CLEARINGHOUSE.;2

changes to: (RECORDS CHBROADCAST DOMAINCACHE ORGCACHE NSNAME)

previous date: 4-Jan-93 17:33:21 {DSK}<python>lde>lispcore>sources>CLEARINGHOUSE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1984, 1985, 1986, 1987, 1988, 1990, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ CLEARINGHOUSECOMS

```
(
    (COURIERPROGRAMS CLEARINGHOUSE)
    (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (SOURCE)
                                             ETHERRECORDS)
      (CONSTANTS (\CH.BROADCAST.SOCKET 20)
                  (\CH.NULL.PROPERTY -1))
      (RECORDS CHBROADCAST DOMAINCACHE ORGCACHE)
      (TEMPLATES CH.PROPERTY)
      (GLOBALVARS LOCAL.CLEARINGHOUSE \CH.CACHE CH.NET.HINT \CH.MAINLOCK NS.SERVER.NAMES.TO.ADDRESSES
                  CLEARINGHOUSE.STRUCTURE.WINDOW CH.PROPERTIES *ASSUME-ZERO-NSSOCKETS*)
      (OPTIMIZERS CH.PROPERTY)
      (ADDVARS (CONSTANTFOLDFNS CH.PROPERTY)))
  (INITVARS (CH.NET.HINT)
            (CH.DEFAULT.DOMAIN NIL)
            (CH.DEFAULT.ORGANIZATION NIL)
            (LOCAL.CLEARINGHOUSE NIL)
            (\CH.CACHE NIL)
            (CLEARINGHOUSE.STRUCTURE.WINDOW NIL)
            (NS.SERVER.NAMES.TO.ADDRESSES NIL)
            (\CH.MAINLOCK (CREATE.MONITORLOCK "Clearinghouse")))
  [P (CL:PROCLAIM ' (CL:SPECIAL CH.DEFAULT.DOMAIN CH.DEFAULT.ORGANIZATION)
    (ADDVARS (\SYSTEMCACHEVARS LOCAL.CLEARINGHOUSE \CH.CACHE NS.SERVER.NAMES.TO.ADDRESSES)
      (CH.PROPERTIES (ALL 0)
        (ALIAS 1)
        (ALIASES 2)
        (MEMBERS 3)
        (ADDRESS.LIST 4)
        (AUTHENTICATION.KEYS 6)
        (AUTHENTICATION.LEVEL 8)
        (MAILPRIMARY 30)
        (MAILBOXES 31)
        (MAILBOX.ACCESSCONTROL 32)
        (SERVICES 51)
        (FILE.SERVICE 10000)
        (PRINT.SERVICE 10001)
        (INTERNET.ROUTING.SERVICE 10002)
        (USER 10003)
        (MAIL.SERVICE 10004)
        (WORKSTATION 10005)
        (EXTERNAL.COMMUNICATION.SERVICE 10006)
        (RS232CPORT 10007)
        (INTERACTIVE.TERMINAL.SERVICE 10008)
        (GATEWAY.SERVICE 10009)
        (IBM3270.HOST 10010)
        (MAIL.GATEWAY 10011)
        (SIEMENS.9750.HOST 10012)
        (ADOBE.SERVICE 10013)
        (LIBRARIAN.SERVICE 10014)
        (TTX.GATEWAY 10015)
        (AUTHENTICATION.SERVICE 10016)
        (REMOTE.BATCH.SERVICE 10017)
        (NETWORK 10018)
        (NETWORK.SERVERS 10019)
        (CIU 10020)
        (CLEARINGHOUSE.SERVICE 10021)
        (USERGROUP 10022)
        (FETCH.SERVICE 10023)
        (SERVER 10024)
        (USERDATA 20000)
        (RS232CDATA 20001)
        (IBM3270HOSTDATA 20002)
        (SIEMENS9750HOSTDATA 20003)
        (CANMAILTO 20005)
        (MAILGATEWAYROUTEDATA 20006)
        (FOREIGNMAILSYSTEMNAME 20007)
        (RS232C.BACK 20102)
        (IBM3270.HOST.BACK 20103)
        (ASSOCIATED.WORKSTATION 30005)
        (FILESERVER 10)))
    (COMS
      ; Clearinghouse names
```

```

(RECORDS NSNAME)
(P (DEFPRINT 'NSNAME (FUNCTION \NSNAME.DEFPRINT)))
(PROP COURIERDEF NSNAME NSNAME2)
(FNS \NSNAME.DEFPRINT NSNAME.TO.STRING COURIER.READ.NSNAME COURIER.WRITE.NSNAME
  COURIER.NSNAME.LENGTH)
(COMS ; Finding Clearinghouse
(FNS GETCLEARINGHOUSE \CH.CHECK.CLEARINGHOUSE START.CLEARINGHOUSE SHOW.CLEARINGHOUSE CH.FINDSERVER
  \CH.FIND.ORG.SERVER \CH.LOCATE.SERVERS \CH.PRINT.BLURB \CH.UPDATE.CACHE EQUAL.CH.NAMES
  PARSE.NSNAME.CH.NAME.TO.STRING CANONICAL.CH.NAME.CH.PROPERTY.CH.GETAUTHENTICATOR)
(FNS CH.SERVERS \CH.GUESS.NEW.PROPERTIES))
(COMS ; Clearinghouse calls
(FNS CH.DOMAINS.SERVED.CH.CREATE.OBJECT.CH.DELETE.OBJECT.CH.CREATE.ALIAS.CH.DELETE.ALIAS
  CH.LIST.ALIASES.CH.LIST.ALIASES.OF.CH.LOOKUP.OBJECT.CH.DELETE.PROPERTY)
(FNS CH.RETRIEVE.MEMBERS.CH.ISMEMBER.CH.ADD.SELF.CH.DELETE.SELF)
(FNS CH.RETRIEVE.ITEM.CH.ADD.ITEM.PROPERTY.CH.CHANGE.ITEM)
(FNS CH.LIST.DOMAINS)
;; NOTE: the following functions were moved to the file AUTHENTICATION temporarily for benefit of Lyric users wanting a functional
;; NSMAINTAIN: CH.LIST.PROPERTIES.CH.LIST.ORGANIZATIONS.CH.LIST.OBJECTS.CH.DELETE.MEMBER
;; CH.ADD.MEMBER.CH.ADD.GROUP.PROPERTY
)
(COMS ; Other entries
(FNS LOOKUP.NS.SERVER \CANONICAL.NSHOSTNAME.CH.CANONICAL.NAME)
(INITVARS (*ASSUME-ZERO-NSSOCKETS* T)))
(FILE AUTHENTICATION))

```

:: Clearinghouse Protocol

(COURIERPROGRAM CLEARINGHOUSE (2 3)

TYPES

```

[ (ORGANIZATION STRING)
  (DOMAIN STRING)
  (OBJECT STRING)
  (ORGANIZATION.NAME ORGANIZATION)
  (DOMAIN.NAME NSNAME2)
  (OBJECT.NAME NSNAME)
  (NAME NSNAME)
  (ORGANIZATION.NAME.PATTERN ORGANIZATION)
  (DOMAIN.NAME.PATTERN NSNAME2)
  (OBJECT.NAME.PATTERN NSNAME)
  (PROPERTY LONGCARDINAL)
  (PROPERTIES (SEQUENCE PROPERTY))
  (ITEM (SEQUENCE UNSPECIFIED))
  (NETWORK.ADDRESS NSADDRESS)
  (NETWORK.ADDRESS.LIST (SEQUENCE NETWORK.ADDRESS))
  [AUTHENTICATOR (RECORD (CREDENTIALS (AUTHENTICATION . CREDENTIALS))
    (VERIFIER (AUTHENTICATION . VERIFIER]
  [MAILBOX.VALUES (RECORD (TIME TIME)
    (MAIL.SERVICE (SEQUENCE NAME])
  (USERDATA.VALUE (RECORD (LAST.NAME.INDEX CARDINAL)
    (FILE.SERVICE NAME)))
  (WHICH.ARGUMENT (ENUMERATION (FIRST 1)
    (SECOND 2)))
  (ARGUMENT.PROBLEM (ENUMERATION (IllegalProperty 10)
    (IllegalOrganization 11)
    (IllegalDomain 12)
    (IllegalObject 13)
    (NoSuchOrganization 14)
    (NoSuchDomain 15)
    (NoSuchObject 16)))
  (CALL.PROBLEM (ENUMERATION (AccessRightsInsufficient 1)
    (TooBusy 2)
    (ServerDown 3)
    (USE.COURIER 4)
    (Other 5)))
  (PROPERTY.PROBLEM (ENUMERATION (Missing 20)
    (WrongType 21)))
  (UPDATE.PROBLEM (ENUMERATION (NoChange 30)
    (OutOfDate 31)
    (ObjectOverflow 32)
    (DatabaseOverflow 33])

```

PROCEDURES

```

((RETRIEVE.ADDRESSES 0 NIL RETURNS (NETWORK.ADDRESS.LIST)
  REPORTS
  (CALL.ERROR))
(LIST.DOMAINS.SERVED 1 (BULK.DATA.SINK AUTHENTICATOR)
  RETURNS NIL REPORTS (CALL.ERROR))
(CREATE.OBJECT 2 (OBJECT.NAME AUTHENTICATOR)
  RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR UPDATE.ERROR WRONG.SERVER))
(DELETE.OBJECT 3 (OBJECT.NAME AUTHENTICATOR)
  RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR UPDATE.ERROR WRONG.SERVER))
(LOOKUP.OBJECT 4 (OBJECT.NAME.PATTERN AUTHENTICATOR)
  RETURNS
  (OBJECT.NAME)
  REPORTS
  (ARGUMENT.ERROR CALL.ERROR WRONG.SERVER))
(LIST.ORGANIZATIONS 5 (ORGANIZATION.NAME.PATTERN BULK.DATA.SINK AUTHENTICATOR)

```

```

    RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(LIST.DOMAINS 6 (DOMAIN.NAME.PATTERN BULK.DATA.SINK AUTHENTICATOR)
    RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(LIST.OBJECTS 7 (OBJECT.NAME.PATTERN PROPERTY BULK.DATA.SINK AUTHENTICATOR)
    RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(LIST.ALIASES 8 (OBJECT.NAME.PATTERN BULK.DATA.SINK AUTHENTICATOR)
    RETURNS NIL REPORTS (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(LIST.ALIASES.OF 9 (OBJECT.NAME.PATTERN BULK.DATA.SINK AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(CREATE.ALIAS 10 (OBJECT.NAME OBJECT.NAME AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR UPDATE.ERROR WRONG.SERVER))
(DELETE.ALIAS 11 (OBJECT.NAME AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR UPDATE.ERROR WRONG.SERVER))
(ADD.GROUP.PROPERTY 12 (OBJECT.NAME PROPERTY BULK.DATA.SOURCE AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(ADD.ITEM.PROPERTY 13 (OBJECT.NAME PROPERTY ITEM AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(DELETE.PROPERTY 14 (OBJECT.NAME PROPERTY AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(LIST.PROPERTIES 15 (OBJECT.NAME.PATTERN AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME PROPERTIES)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR WRONG.SERVER))
(RETRIEVE.ITEM 16 (OBJECT.NAME.PATTERN PROPERTY AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME ITEM)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR WRONG.SERVER))
(CHANGE.ITEM 17 (OBJECT.NAME PROPERTY ITEM AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(RETRIEVE.MEMBERS 18 (OBJECT.NAME.PATTERN PROPERTY BULK.DATA.SINK AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR WRONG.SERVER))
(ADD.MEMBER 19 (OBJECT.NAME PROPERTY NSNAME AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(ADD.SELF 20 (OBJECT.NAME PROPERTY AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(DELETE.MEMBER 21 (OBJECT.NAME PROPERTY NSNAME AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(DELETE.SELF 22 (OBJECT.NAME PROPERTY AUTHENTICATOR)
    RETURNS
    (OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR UPDATE.ERROR WRONG.SERVER))
(IS.MEMBER 23 (OBJECT.NAME.PATTERN PROPERTY PROPERTY NSNAME AUTHENTICATOR)
    RETURNS
    (BOOLEAN OBJECT.NAME)
    REPORTS
    (ARGUMENT.ERROR AUTHENTICATION.ERROR CALL.ERROR PROPERTY.ERROR WRONG.SERVER))

```

ERRORS

```

[ (CALL.ERROR 1 (CALL.PROBLEM))
  (ARGUMENT.ERROR 2 (ARGUMENT.PROBLEM WHICH.ARGUMENT))
  (PROPERTY.ERROR 3 (PROPERTY.PROBLEM OBJECT.NAME))
  (UPDATE.ERROR 4 (UPDATE.PROBLEM BOOLEAN WHICH.ARGUMENT OBJECT.NAME))

```

(WRONG.SERVER 5 (OBJECT.NAME))
(AUTHENTICATION.ERROR 6 ((AUTHENTICATION . PROBLEM]))

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (SOURCE)
ETHERRECORDS)

(DECLARE%: EVAL@COMPILE

(RPAQQ \CH.BROADCAST.SOCKET 20)

(RPAQQ \CH.NULL.PROPERTY -1)

(CONSTANTS (\CH.BROADCAST.SOCKET 20)
(\CH.NULL.PROPERTY -1))
)

(DECLARE%: EVAL@COMPILE

[ACCESSFNS CHBROADCAST ((CHBROADCASTBODY (fetch (PACKETEXCHANGEXIP PACKETEXCHANGE) of DATUM)))
(BLOCKRECORD CHBROADCASTBODY ((LOW.VERSION WORD)
(HIGH.VERSION WORD)
(MSGTYPE WORD)
(ZERO2 WORD)
(PROGRAM# FIXP)
(VERSION# WORD)
(ZERO3 WORD))

(RECORD DOMAINCACHE (DCDOMAIN DCKNOWNSERVERS DCHINTSERVERS DCCOMPLETE))

(RECORD ORGCACHE (OCORGANIZATION OCALLSERVERS . OCDOMAINS))
)

(SETTEMPLATE 'CH.PROPERTY ' (PROP))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS LOCAL.CLEARINGHOUSE \CH.CACHE CH.NET.HINT \CH.MAINLOCK NS.SERVER.NAMES.TO.ADDRESSES
CLEARINGHOUSE.STRUCTURE.WINDOW CH.PROPERTIES *ASSUME-ZERO-NSSOCKETS*)
)

(DEFOPTIMIZER CH.PROPERTY (&WHOLE FORM) ; for some reason, compiler needs more than having
; CH.PROPERTY on CONSTANTFOLDFNS
(OR (CAR (CONSTANTEXPRESSIONP FORM))
' IGNOREMACRO))

(ADDTOVAR CONSTANTFOLDFNS CH.PROPERTY)
)

(RPAQ? CH.NET.HINT)

(RPAQ? CH.DEFAULT.DOMAIN NIL)

(RPAQ? CH.DEFAULT.ORGANIZATION NIL)

(RPAQ? LOCAL.CLEARINGHOUSE NIL)

(RPAQ? \CH.CACHE NIL)

(RPAQ? CLEARINGHOUSE.STRUCTURE.WINDOW NIL)

(RPAQ? NS.SERVER.NAMES.TO.ADDRESSES NIL)

(RPAQ? \CH.MAINLOCK (CREATE.MONITORLOCK "Clearinghouse"))

(CL:PROCLAIM '(CL:SPECIAL CH.DEFAULT.DOMAIN CH.DEFAULT.ORGANIZATION))

(ADDTOVAR \SYSTEMCACHEVARS LOCAL.CLEARINGHOUSE \CH.CACHE NS.SERVER.NAMES.TO.ADDRESSES)

(ADDTOVAR CH.PROPERTIES
(ALL 0)
(ALIAS 1)
(ALIASES 2)
(MEMBERS 3)
(ADDRESS.LIST 4)
(AUTHENTICATION.KEYS 6)
(AUTHENTICATION.LEVEL 8)
(MAILPRIMARY 30)
(MAILBOXES 31)
(MAILBOX.ACCESSCONTROL 32)
(SERVICES 51)
(FILE.SERVICE 10000)
(PRINT.SERVICE 10001)
(INTERNET.ROUTING.SERVICE 10002)
(USER 10003)

```
(MAIL.SERVICE 10004)
(WORKSTATION 10005)
(EXTERNAL.COMMUNICATION.SERVICE 10006)
(RS232CPORT 10007)
(INTERACTIVE.TERMINAL.SERVICE 10008)
(GATEWAY.SERVICE 10009)
(IBM3270.HOST 10010)
(MAIL.GATEWAY 10011)
(SIEMENS.9750.HOST 10012)
(ADOBE.SERVICE 10013)
(LIBRARIAN.SERVICE 10014)
(TTX.GATEWAY 10015)
(AUTHENTICATION.SERVICE 10016)
(REMOTE.BATCH.SERVICE 10017)
(NETWORK 10018)
(NETWORK.SERVERS 10019)
(CIU 10020)
(CLEARINGHOUSE.SERVICE 10021)
(USERGROUP 10022)
(FETCH.SERVICE 10023)
(SERVER 10024)
(USERDATA 20000)
(RS232CDATA 20001)
(IBM3270HOSTDATA 20002)
(SIEMENS9750HOSTDATA 20003)
(CANMAILTO 20005)
(MAILGATEWAYROUTEDATA 20006)
(FOREIGNMAILSYSTEMNAME 20007)
(RS232C.BACK 20102)
(IBM3270.HOST.BACK 20103)
(ASSOCIATED.WORKSTATION 30005)
(FILESERVER 10)
```

:: Clearinghouse names

```
(DECLARE%: EVAL@COMPILE
(DATATYPE NSNAME ((NSOBJECT POINTER)
                  (NSDOMAIN POINTER)
                  (NSORGANIZATION POINTER))
              (* Canonical three-part Clearinghouse name)
)
)
```

```
(/DECLAREDATATYPE 'NSNAME' (POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 6)
```

```
(DEFPRINT 'NSNAME (FUNCTION \NSNAME.DEFPRINT))
(PUTPROPS NSNAME COURIERDEF (COURIER.READ.NSNAME COURIER.WRITE.NSNAME COURIER.NSNAME.LENGTH))
(PUTPROPS NSNAME2 COURIERDEF (COURIER.READ.NSNAME COURIER.WRITE.NSNAME))
(DEFINEQ
```

```
(\NSNAME.DEFPRINT
[LAMBDA (NAME STREAM)
(LIST (NSNAME.TO.STRING NAME))]
(* bvm%: "28-Jun-84 16:36")
```

```
(\NSNAME.TO.STRING
[LAMBDA (NSNAME FULLNAMEFLG)
(PROG ([OBJ (ffetch NSOBJECT of (SETQ NSNAME (\DTEST NSNAME 'NSNAME)
(DOM (ffetch NSDOMAIN of NSNAME))
(ORG (ffetch NSORGANIZATION of NSNAME))
(COLON ":"))
(RETURN (COND
((NOT OBJ) ; Two-part name
(CONCAT DOM COLON ORG))
((OR FULLNAMEFLG (NOT (STRING-EQUAL ORG CH.DEFAULT.ORGANIZATION)))
; No defaults
(CONCAT OBJ COLON DOM COLON ORG))
(T (CONCAT OBJ COLON (COND
((STRING-EQUAL DOM CH.DEFAULT.DOMAIN)
"")
(T DOM]))
```

```
(\COURIER.READ.NSNAME
[LAMBDA (STREAM PROGRAM TYPE)
(* bvm%: "27-Jun-84 15:41")
```

::: Read a Clearinghouse name: 3 strings, or 2 strings if reading a 2-part name

```
(create NSNAME
NSORGANIZATION _ (COURIER.READ.STRING STREAM)
NSDOMAIN _ (COURIER.READ.STRING STREAM)
```

NSOBJECT _ (AND (EQ TYPE 'NSNAME)
(COURIER.READ.STRING STREAM])

(COURIER.WRITE.NSNAME

[LAMBDA (STREAM NAME PROGRAM TYPE) (* bvm%: "27-Jun-84 15:43")
[COURIER.WRITE.STRING STREAM (fetch NSORGANIZATION of (\DTEST NAME 'NSNAME)
(COURIER.WRITE.STRING STREAM (fetch NSDOMAIN of NAME))
(COND
((EQ TYPE 'NSNAME) ; full 3-part name
(COURIER.WRITE.STRING STREAM (fetch NSOBJECT of NAME])

(COURIER.NSNAME.LENGTH

[LAMBDA (NSNAME PROGRAM TYPE) ; Edited 21-Jul-87 17:00 by bvm:
;; Return the representation length of an NSNAME, or NIL if we can't cheaply.

(PROG NIL
(RETURN (+ [COND
((EQ TYPE 'NSNAME2)
0)
(T (OR (COURIER.REP.LENGTH (fetch NSOBJECT of NSNAME)
NIL
'STRING)
(RETURN NIL])
(OR (COURIER.REP.LENGTH (fetch NSDOMAIN of NSNAME)
NIL
'STRING)
(RETURN NIL])
(OR (COURIER.REP.LENGTH (fetch NSORGANIZATION of NSNAME)
NIL
'STRING)
(RETURN NIL])

;; Finding Clearinghouse

(DEFINEQ

(GETCLEARINGHOUSE

[LAMBDA NIL ; Edited 24-Jul-87 17:33 by bvm:
(COND
((AND CH.DEFAULT.DOMAIN CH.DEFAULT.ORGANIZATION LOCAL.CLEARINGHOUSE))
(T (WITH.MONITOR \CH.MAINLOCK
[OR LOCAL.CLEARINGHOUSE (repeatuntil (SETQ \$\$VAL (COURIER.BROADCAST.CALL \CH.BROADCAST.SOCKET
' CLEARINGHOUSE
' RETRIEVE.ADDRESSES NIL
(FUNCTION \CH.CHECK.CLEARINGHOUSE)
CH.NET.HINT "Clearinghouse servers"])]))

(\CH.CHECK.CLEARINGHOUSE

[LAMBDA (ADDRESSES) (* bvm%: "15-Aug-84 12:49")
(PROG ((ADDR (BESTNSADDRESS ADDRESSES T))
DOMAINS DEFAULT)
(RETURN (COND

((AND ADDR (SETQ DOMAINS (CH.DOMAINS.SERVED ADDR))
(find old DEFAULT in DOMAINS suchthat (PROGN
; Skip over the stupid domain
(NOT (STREQUAL (fetch NSORGANIZATION
of DEFAULT)
"..."]
(COND
((OR (NULL CH.DEFAULT.DOMAIN)
(NULL CH.DEFAULT.ORGANIZATION)) ; Use the first domain that this server serves to set the default
; domain and organization.
; Skip over the stupid domain
(printout PROMPTWINDOW T "[Default Clearinghouse domain set to "
(SETQ CH.DEFAULT.DOMAIN (fetch NSDOMAIN of DEFAULT))
": "
(SETQ CH.DEFAULT.ORGANIZATION (fetch NSORGANIZATION of DEFAULT))
"]"))))
(\CH.UPDATE.CACHE (create NSNAME
NSOBJECT _ (MKSTRING ADDR)
NSDOMAIN _ "CHServers"
NSORGANIZATION _ "CHServers")
ADDR DOMAINS)
(SETQ LOCAL.CLEARINGHOUSE ADDR])

(START.CLEARINGHOUSE

[LAMBDA (RESTARTFLG) (* bvm%: "27-Jun-84 10:57")
(COND
((OR RESTARTFLG (NULL LOCAL.CLEARINGHOUSE))
(SETQ \NS.ROUTING.TABLE.RADIUS 5)
(SETQ NS.SERVER.NAMES.TO.ADDRESSES (SETQ \CH.CACHE (SETQ LOCAL.CLEARINGHOUSE NIL)))

(GETCLEARINGHOUSE))
LOCAL.CLEARINGHOUSE])

(SHOW.CLEARINGHOUSE

[LAMBDA (ENTIRE.CLEARINGHOUSE? DONT.GRAPH) ; Edited 23-Jul-87 12:02 by bvm:
(PROG (SEXPR)
[SETQ SEXPR (CONS "" (COND
[ENTIRE.CLEARINGHOUSE? ; Find all domains in all organizations.
(for ORG in (CH.LIST.ORGANIZATIONS ""))
collect (CONS ORG (CH.LIST.DOMAINS (CONCAT ":" ORG)
(T ; Use cached structure.
(for ORG in \CH.CACHE collect (CONS (fetch OCORGANIZATION of ORG)
(for DOM in (fetch OCDOMAINS of ORG)
collect (fetch DCDOMAIN of DOM]
(COND
(DONT.GRAPH (RETURN SEXPR)))
(FILESLOAD (SYSLOAD FROM VALUEOF LISPUSERSDIRECTORIES)
GRAPHER)
(SETQ CLEARINGHOUSE.STRUCTURE.WINDOW (SHOWGRAPH (LAYOUTSEXPR SEXPR 'HORIZONTAL NIL
' (HELVETICA 10 BOLD))
(OR CLEARINGHOUSE.STRUCTURE.WINDOW "Clearinghouse
structure")))
[WINDOWPROP CLEARINGHOUSE.STRUCTURE.WINDOW 'CLOSEFN (FUNCTION (LAMBDA NIL
(SETQ CLEARINGHOUSE.STRUCTURE.WINDOW
NIL]
(RETURN CLEARINGHOUSE.STRUCTURE.WINDOW])

(CH.FINDSERVER

[LAMBDA (DOMAINPATTERN NOERRORFLG DONTPROBEFLG) ; Edited 23-Jul-87 12:02 by bvm:
;; Find a Clearinghouse which serves the specified domain and return its NS address. If DONTPROBEFLG is T, just search the cache.
(OR (type? NSNAME DOMAINPATTERN)
(SETQ DOMAINPATTERN (PARSE.NSNAME DOMAINPATTERN 2)))
(LET ((ORGANIZATION (fetch NSORGANIZATION of DOMAINPATTERN))
(DOMAIN (fetch NSDOMAIN of DOMAINPATTERN))
(ORGANIZATION.INFO)
(SETQ ORGANIZATION.INFO (\CH.FIND.ORG.SERVER ORGANIZATION NOERRORFLG DONTPROBEFLG))
(COND
[(STRING-EQUAL DOMAIN "") ; Any server in the org will do.
(CAR (CAR (fetch OCALLSERVERS of ORGANIZATION.INFO)
[(for DOMAIN.INFO in (fetch OCDOMAINS of ORGANIZATION.INFO) when (STRING-EQUAL (fetch DCDOMAIN
of DOMAIN.INFO)
DOMAIN)
do (RETURN (CAR (CAR (fetch DCKNOWNSERVERS of DOMAIN.INFO)
(DONTPROBEFLG (AND (NOT NOERRORFLG)
(ERROR "Couldn't find Clearinghouse server for domain" DOMAINPATTERN T)))
(T ;; Ask a clearinghouse in ORGANIZATION to find servers for this domain. For simplicity, assume the first one will tell us. This
;; should be 'Local Clearinghouse' if it serves ORGANIZATION
(\CH.LOCATE.SERVERS (CAR (CAR (fetch OCALLSERVERS of ORGANIZATION.INFO)))
(create NSNAME
NSOBJECT _ DOMAIN
NSDOMAIN _ ORGANIZATION
NSORGANIZATION _ "CHServers")
NOERRORFLG ORGANIZATION DOMAIN)
(CH.FINDSERVER DOMAINPATTERN NOERRORFLG T])

(\CH.FIND.ORG.SERVER

[LAMBDA (ORGANIZATION NOERRORFLG DONTPROBEFLG) ; Edited 23-Jul-87 12:19 by bvm:
(COND
((find ORGINFO in \CH.CACHE suchthat (STRING-EQUAL (fetch OCORGANIZATION of ORGINFO)
ORGANIZATION)))
(DONTPROBEFLG (AND (NOT NOERRORFLG)
(ERROR "Couldn't find Clearinghouse server for organization" ORGANIZATION T)))
((STRING-EQUAL ORGANIZATION "CHServers")
;; Everyone handles this org, so create a fake organization "CHServers" whose domains are "CHServers" (served by everybody) and each
;; known organization (served by all servers for that org).
(GETCLEARINGHOUSE)
[LET (ALLSERVERS KNOWNDOMAINS)
[SETQ KNOWNDOMAINS (for ORG in \CH.CACHE collect (SETQ ALLSERVERS (APPEND (fetch OCALLSERVERS
of ORG)
ALLSERVERS))
; Note servers for grand list
(create DOMAINCACHE
DCDOMAIN _ (fetch OCORGANIZATION of ORG)
DCKNOWNSERVERS _ (fetch OCALLSERVERS
of ORG]
(SETQ ALLSERVERS (SORT.NSADDRESSES.BY.DISTANCE ALLSERVERS))
(SETQ \CH.CACHE (NCONC1 \CH.CACHE (create ORGCACHE
OCORGANIZATION _ ORGANIZATION
OCALLSERVERS _ ALLSERVERS
OCDOMAINS _
(CONS (create DOMAINCACHE

```

DCDOMAIN _ ORGANIZATION
DCKNOWNSERVERS _ ALLSERVERS)
KNOWNDOMAINS]
(\CH.FIND.ORG.SERVER ORGANIZATION NOERRORFLG T)
(T ; Search for servers for this org by asking for
; org:CHServers:CHServers
(\CH.LOCATE.SERVERS (GETCLEARINGHOUSE)
(create NSNAME
NSOBJECT _ ORGANIZATION
NSDOMAIN _ "CHServers"
NSORGANIZATION _ "CHServers")
NOERRORFLG ORGANIZATION)
(\CH.FIND.ORG.SERVER ORGANIZATION NOERRORFLG T))

```

(\CH.LOCATE.SERVERS

```
[LAMBDA (CHSERVER SERVERGROUP NOERRORFLG ORGANIZATION DOMAIN) (* bvm%: "26-Jul-85 12:42")
```

;;; Talks to clearinghouse CHSERVER asking it to enumerate the class of servers SERVERGROUP. We then locate each server and update the cache

```

(PROG (STREAM CHLIST CHNAMES NOROUTE)
(\CH.PRINT.BLURB "Finding Clearinghouse server for " DOMAIN ORGANIZATION)
(OR [COND
((SETQ STREAM (COURIER.OPEN CHSERVER NIL NOERRORFLG 'CLEARINGHOUSE))
(RESETLST
(RESETSAVE NIL (LIST (FUNCTION \SPP.RESETCLOSE)
STREAM))
[COND
((AND (SETQ CHNAMES (CH.RETRIEVE.MEMBERS SERVERGROUP (CH.PROPERTY 'MEMBERS)
STREAM))
(NEQ (CAR CHNAMES)
'ERROR))
(SETQ CHLIST (SORT.NSADDRESSES.BY.DISTANCE
(for CH in CHNAMES bind INFO when (SETQ INFO (COURIER.CALL
STREAM
' CLEARINGHOUSE
'RETRIEVE.ITEM CH
(CH.PROPERTY
' ADDRESS.LIST)
(CH.GETAUTHENTICATOR)
' NOERROR))
collect (LIST (CAR (COURIER.READ.REP (CADR INFO)
' CLEARINGHOUSE
' NETWORK.ADDRESS.LIST))
(CAR INFO))

```

;; Now have a list of all servers for the desired domain in hop order. Find the first one that's up, and make sure it serves the domain that the hint claimed. Could update cache with all domains it serves, but then we would be in danger of picking a distant server we know about in preference to a close server we haven't discovered yet

```

(for PAIR in CHLIST bind CHDOMAINS DOM
when [COND
([NOT (\LOCATE.NSNET (fetch NSNET of (CAR PAIR)
; Give up when we get to inaccessible hosts
(SETQ NOROUTE T)
(RETURN))
((SETQ CHDOMAINS (CH.DOMAINS.SERVED (CAR PAIR)))
(for old DOM in CHDOMAINS thereis (AND (OR (NULL DOMAIN)
(String-EQUAL (fetch NSDOMAIN of DOM)
DOMAIN)
(String-EQUAL (fetch NSORGANIZATION
of DOM)
ORGANIZATION])
do (\CH.UPDATE.CACHE (CADR PAIR)
(CAR PAIR)
(LIST DOM))
(RETURN T]
(\CH.PRINT.BLURB (COND
((EQ (CAR CHNAMES)
'ERROR)
"Unknown domain: ")
(NOROUTE "No network route to Clearinghouse for ")
(T "Unable to contact Clearinghouse for "))
DOMAIN ORGANIZATION])

```

(\CH.PRINT.BLURB

```
[LAMBDA (MSG DOMAIN ORGANIZATION) (* bvm%: "29-Jul-84 22:40")
(printout PROMPTWINDOW T "[" MSG)
[COND
(DOMAIN (printout PROMPTWINDOW DOMAIN '%:]
(printout PROMPTWINDOW ORGANIZATION "]""))

```

(\CH.UPDATE.CACHE

```
[LAMBDA (OBJECT ADDRESS DOMAINS.SERVED) ; Edited 23-Jul-87 12:19 by bvm:
```

;; Note the clearinghouse named OBJECT with given ADDRESS as serving each of the domains in DOMAINS.SERVED.

;; The Clearinghouse cache is sorted by organization and then by domain within org. Currently, we don't time out entries or do anything useful
;; when servers go down.

```
(PROG (NAME.AND.ADDRESS ORGANIZATION DOMAINSTRING ORGANIZATION.INFO)
  (printout PROMPTWINDOW .TAB0 0 "[Noting Clearinghouse " (fetch NSOBJECT of OBJECT)
    "]"")
  (SETQ NAME.AND.ADDRESS (LIST ADDRESS OBJECT))
  [for DOMAIN in DOMAINS.SERVED do (SETQ ORGANIZATION (fetch NSORGANIZATION of DOMAIN))
    (SETQ DOMAINSTRING (fetch NSDOMAIN of DOMAIN))
    [COND
      ([NOT (SETQ ORGANIZATION.INFO
        (find X in \CH.CACHE suchthat (STRING-EQUAL (fetch
          OCORGANIZATION
            of X)
          ORGANIZATION])
        (SETQ \CH.CACHE (NCONC1 \CH.CACHE (SETQ ORGANIZATION.INFO
          (create ORGCACHE
            OCORGANIZATION _
            ORGANIZATION)
          [for DOMAIN.INFO in (fetch OCDOMAINS of ORGANIZATION.INFO)
            when (STRING-EQUAL (fetch DCDOMAIN of DOMAIN.INFO)
              DOMAINSTRING)
            do (push (fetch DCKNOWNSERVERS of DOMAIN.INFO)
              NAME.AND.ADDRESS)
              (RETURN (SORT.NSADDRESSES.BY.DISTANCE (fetch DCKNOWNSERVERS
                of DOMAIN.INFO)))
            finally (push (fetch OCDOMAINS of ORGANIZATION.INFO)
              (create DOMAINCACHE
                DCDOMAIN _ DOMAINSTRING
                DCKNOWNSERVERS _ (LIST NAME.AND.ADDRESS]
              (if (NOT (MEMB NAME.AND.ADDRESS (fetch OCALLSERVERS of ORGANIZATION.INFO)
                ))
                then ; Also add server to list of all servers for the organization. Note
                  ; that the MEMB test is a quicky, only testing for duplicates on
                  ; this call.
                  (SORT.NSADDRESSES.BY.DISTANCE (push (fetch OCALLSERVERS
                    of ORGANIZATION.INFO)
                    NAME.AND.ADDRESS]
                ))
            ))
          ))
    ]
  (COND
    (CLEARINGHOUSE.STRUCTURE.WINDOW (SHOW.CLEARINGHOUSE])
```

(EQUAL.CH.NAMES

[LAMBDA (NAME1 NAME2)

; Edited 6-Dec-88 16:22 by jds

;; Check if two Clearinghouse names are the same.

```
(AND (type? NSNAME NAME1)
  (type? NSNAME NAME2)
  (STRING-EQUAL (fetch (NSNAME NSOBJECT) of NAME1)
    (fetch (NSNAME NSOBJECT) of NAME2))
  (STRING-EQUAL (fetch (NSNAME NSDOMAIN) of NAME1)
    (fetch (NSNAME NSDOMAIN) of NAME2))
  (STRING-EQUAL (fetch (NSNAME NSORGANIZATION) of NAME1)
    (fetch (NSNAME NSORGANIZATION) of NAME2]))
```

(PARSE.NSNAME

[LAMBDA (NAME %#PARTS DEFAULTDOMAIN)

(* bvm%: " 3-Jul-84 16:26")

;;; Coerces NAME to a 3-part NS name. If #PARTS is 1 or 2 then interprets NAME as 1 or 2-part name, discarding the object and/or domain name if
;;; supplied. 3-part names are objects of type NSNAME --- 2-part names are objects of type NSNAME with an OBJECT field of NIL --- A 1 part name is
;;; simply the organization string. --- DEFAULTDOMAIN is the default to supply for missing parts, which in turn default to CH.DEFAULT.DOMAIN and
;;; CH.DEFAULT.ORGANIZATION --- If NAME is already an NSNAME, returns it or a copy if #PARTS implies coercion.

```
(COND
  ((type? NSNAME NAME)
    (SELECTQ %#PARTS
      (NIL 3)
      NAME)
    (2 (COND
      ((fetch NSOBJECT of NAME)
        (create NSNAME using NAME NSOBJECT _ NIL))
      (T NAME)))
      (1 (fetch NSORGANIZATION of NAME))
      (LISPERROR "ILLEGAL ARG" %#PARTS)))
  [[OR (STRINGP NAME)
    (AND (NULL NAME)
      (SETQ NAME ""))
    (AND (LITATOM NAME)
      (SETQ NAME (MKSTRING NAME]
  (PROG (FIRSTPART SECONDPART THIRDPART I J)
    (GETCLEARINGHOUSE)
    (COND
      [(SETQ I (STRPOS ":" NAME))
        (SETQ FIRSTPART (SUBSTRING NAME 1 (SUB1 I)))
        (SETQ SECONDPART (SUBSTRING NAME (ADD1 I)
          (COND
            ((SETQ J (STRPOS ":" NAME (ADD1 I)))
```

```

                (SETQ THIRDPART (SUBSTRING NAME (ADD1 J)
                NIL))
                (SUB1 J]
    (T (SETQ FIRSTPART NAME)))
    (RETURN (SELECTQ %#PARTS
            (NIL 3)
            (create NSNAME
                    NSOBJECT _ FIRSTPART
                    NSDOMAIN _ (COND
                                (SECONDPART)
                                (DEFAULTDOMAIN (fetch NSDOMAIN of DEFAULTDOMAIN))
                                (T CH.DEFAULT.DOMAIN))
                    NSORGANIZATION _ (COND
                                        (THIRDPART)
                                        (DEFAULTDOMAIN (fetch NSORGANIZATION of DEFAULTDOMAIN))
                                        (T CH.DEFAULT.ORGANIZATION))))
            (2 (create NSNAME
                    NSDOMAIN _ (OR (AND THIRDPART SECONDPART)
                                    FIRSTPART)
                    NSORGANIZATION _ (COND
                                        (THIRDPART)
                                        (SECONDPART)
                                        (DEFAULTDOMAIN (fetch NSORGANIZATION of DEFAULTDOMAIN))
                                        (T CH.DEFAULT.ORGANIZATION))))
            (1 (OR THIRDPART SECONDPART FIRSTPART)))
            (LISPERROR "ILLEGAL ARG" %#PARTS]
    (T (LISPERROR "ILLEGAL ARG" NAME])

```

(CH.NAME.TO.STRING

```

[LAMBDA (NSNAME FULLNAMEFLG)
    (* bvm%: "28-Jun-84 16:37")
    ;; Return a string for a Clearinghouse name. Leaves off default components unless FULLNAMEFLG is set.
    (NSNAME.TO.STRING (COND
                      ((type? NSNAME NSNAME)
                       NSNAME)
                      (T (PARSE.NSNAME NSNAME)))
                      FULLNAMEFLG])

```

(CANONICAL.CH.NAME

```

[LAMBDA (NAME)
    (MKATOM (NSNAME.TO.STRING (PARSE.NSNAME NAME)))
    (* bvm%: "28-Jun-84 16:37")

```

(CH.PROPERTY

```

[LAMBDA (PROP)
    (* bvm%: "27-Aug-86 10:38")
    ;; Return the official Clearinghouse property ID for the specified property.
    (OR (COND
        [(LITATOM PROP)
         (for PAIR in CH.PROPERTIES when (EQ (CAR PAIR)
                                             PROP)
         do (RETURN (CADR PAIR)
                    (T (FIXP PROP)))
         (ERROR "Unknown Clearinghouse property" PROP])

```

(CH.GETAUTHENTICATOR

```

[LAMBDA (NONTRIVIAL)
    (COND
     [NONTRIVIAL (PROG [(INFO (\INTERNAL/GETPASSWORD ' |NS::|)
                            (RETURN (COND
                                    (INFO (SETQ INFO (NS.MAKE.SIMPLE.CREDENTIALS INFO))
                                            (COURIER.CREATE (CLEARINGHOUSE . AUTHENTICATOR)
                                                            CREDENTIALS _ (CAR INFO)
                                                            VERIFIER _ (CDR INFO)))
                                    (T (ERROR!])
                                (T (COURIER.CREATE (CLEARINGHOUSE . AUTHENTICATOR)
                                                    CREDENTIALS _ (COURIER.CREATE (AUTHENTICATION . CREDENTIALS)
                                                                    TYPE _ 'SIMPLE VALUE _ NIL)
                                                    VERIFIER _ ' (0))

```

(DEFINEQ

(CH.SERVERS

```

[LAMBDA NIL
    (* bvm%: "3-Jul-84 14:10")
    ;; Return a list of the names of all Clearinghouse servers.
    (COURIER.CALL (GETCLEARINGHOUSE)
                  ' CLEARINGHOUSE
                  ' RETRIEVE.MEMBERS
                  (create NSNAME
                        NSOBJECT _ "CHServers"
                        NSDOMAIN _ "CHServers"

```

```

      NSORGANIZATION _ "CHServers")
    (CH.PROPERTY ' MEMBERS)
    ' (CLEARINGHOUSE . OBJECT.NAME)
    (CH.GETAUTHENTICATOR])

```

(CH.GUESS.NEW.PROPERTIES

```

[LAMBDA (DOMAIN MINPROPERTYID MAXPROPERTYID) (* bvm%: " 4-Jul-84 14:25")
  ;; This is a hack that finds all the objects in the given domain with any properties in the given range. Useful for finding out what the Services people
  ;; are up to.
  (PROG ((PATTERN (PARSE.NSNAME DOMAIN 2))
    OBJECTS)
    (SETQ PATTERN (create NSNAME using PATTERN NSOBJECT _ "*"))
    (SETQ MINPROPERTYID (OR MINPROPERTYID 1))
    (SETQ MAXPROPERTYID (OR MAXPROPERTYID 25))
    (RETURN (for ID from MINPROPERTYID to MAXPROPERTYID when (SETQ OBJECTS (CH.LIST.OBJECTS PATTERN ID))
      collect (CONS ID OBJECTS]))
  )

```

;; Clearinghouse calls

(DEFINEQ

(CH.DOMAINS.SERVED

```

[LAMBDA (CHADDRESS) (* bvm%: " 3-Jul-84 14:12")
  ;; We wrap this in an NLSETQ because we might get an error underneath the Bulk Data transfer if we're not really talking to a Clearinghouse.
  (CAR (NLSETQ (COURIER.CALL CHADDRESS 'CLEARINGHOUSE 'LIST.DOMAINS.SERVED ' (CLEARINGHOUSE . DOMAIN.NAME)
    (CH.GETAUTHENTICATOR])

```

(CH.CREATE.OBJECT

```

[LAMBDA (OBJECTNAME STREAM) (* jwo%: " 8-Jul-85 14:13")

```

;;; Creates an object with given name, no properties

```

  (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME))
  (OR (COURIER.CALL (OR STREAM (CH.FINDSERVER OBJECTNAME))
    ' CLEARINGHOUSE
    ' CREATE.OBJECT OBJECTNAME (CH.GETAUTHENTICATOR T)
    ' RETURNERRORS)
    OBJECTNAME])

```

(CH.DELETE.OBJECT

```

[LAMBDA (OBJECTNAME STREAM) (* jwo%: " 8-Jul-85 14:06")

```

;;; Deletes specified object from Clearinghouse database

```

  (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME))
  (OR (COURIER.CALL (OR STREAM (CH.FINDSERVER OBJECTNAME))
    ' CLEARINGHOUSE
    ' DELETE.OBJECT OBJECTNAME (CH.GETAUTHENTICATOR T)
    ' RETURNERRORS)
    OBJECTNAME])

```

(CH.CREATE.ALIAS

```

[LAMBDA (ALIAS OBJECTNAME) ; Edited 31-Jul-87 11:02 by bvm:

```

;;; Makes ALIAS be an alias of OBJECTNAME in the Clearinghouse database

```

  (COURIER.CALL (CH.FINDSERVER (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME)))
    ' CLEARINGHOUSE
    ' CREATE.ALIAS
    (PARSE.NSNAME ALIAS)
    OBJECTNAME
    (CH.GETAUTHENTICATOR T)
    ' RETURNERRORS])

```

(CH.DELETE.ALIAS

```

[LAMBDA (ALIAS) (* jwo%: "10-Jun-85 21:03")

```

;;; Deletes specified alias from Clearinghouse database

```

  (COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ ALIAS (PARSE.NSNAME ALIAS)))
    \CH.BROADCAST.SOCKET
    ' CLEARINGHOUSE
    ' DELETE.ALIAS ALIAS (CH.GETAUTHENTICATOR T)
    ' RETURNERRORS])

```

(CH.LIST.ALIASES

```

[LAMBDA (OBJECTNAMEPATTERN) ; Edited 24-Jul-87 17:45 by bvm:

```

```
(COURIER.CALL (CH.FINDSERVER (SETQ OBJECTNAMEPATTERN (PARSE.NSNAME OBJECTNAMEPATTERN)))
' CLEARINGHOUSE
' LIST.ALIASES OBJECTNAMEPATTERN ' (CLEARINGHOUSE . OBJECT)
(CH.GETAUTHENTICATOR)
' RETURNERRORS])
```

(CH.LIST.ALIASES.OF

```
[LAMBDA (OBJECTPATTERN) ; Edited 24-Jul-87 18:05 by bvm:
(COURIER.CALL (CH.FINDSERVER (SETQ OBJECTPATTERN (PARSE.NSNAME OBJECTPATTERN)))
' CLEARINGHOUSE
' LIST.ALIASES.OF OBJECTPATTERN ' (CLEARINGHOUSE . OBJECT.NAME)
(CH.GETAUTHENTICATOR)
' NOERROR])
```

(CH.LOOKUP.OBJECT

```
[LAMBDA (OBJECTPATTERN) (* bvm%: "28-Jun-84 16:37")
```

::: Returns the canonical name of the specified object. If object contains wildcards, result is the first match

```
(SETQ OBJECTPATTERN (PARSE.NSNAME OBJECTPATTERN))
(PROG ((ADDRESS (CH.FINDSERVER OBJECTPATTERN T)))
(RETURN (AND ADDRESS (COURIER.EXPEDITED.CALL ADDRESS \CH.BROADCAST.SOCKET 'CLEARINGHOUSE
' LOOKUP.OBJECT OBJECTPATTERN (CH.GETAUTHENTICATOR)
' NOERROR]))
```

(CH.DELETE.PROPERTY

```
[LAMBDA (OBJECTNAME PROPERTY) (* jwo%: "10-Jun-85 21:04")
(COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME)))
\CH.BROADCAST.SOCKET
' CLEARINGHOUSE
' DELETE.PROPERTY OBJECTNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(CH.GETAUTHENTICATOR T)
' RETURNERRORS])
```

(DEFINEQ

(CH.RETRIEVE.MEMBERS

```
[LAMBDA (OBJECTPATTERN PROPERTY STREAM) (* bvm%: "21-Feb-86 14:33")
(SETQ OBJECTPATTERN (PARSE.NSNAME OBJECTPATTERN))
(COURIER.CALL (OR STREAM (CH.FINDSERVER OBJECTPATTERN))
' CLEARINGHOUSE
' RETRIEVE.MEMBERS OBJECTPATTERN [COND
(PROPERTY (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY)))
(T (CH.PROPERTY 'MEMBERS]
' (CLEARINGHOUSE . OBJECT.NAME)
(CH.GETAUTHENTICATOR)
' RETURNERRORS])
```

(CH.ISMEMBER

```
[LAMBDA (GROUPNAME PROPERTY SECONDARYPROPERTY NAME) (* bvm%: "4-Jul-84 14:21")
(CAR (COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ GROUPNAME (PARSE.NSNAME GROUPNAME)))
\CH.BROADCAST.SOCKET
' CLEARINGHOUSE
' IS.MEMBER GROUPNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(COND
((NULL SECONDARYPROPERTY)
\CH.NULL.PROPERTY)
((FIXP SECONDARYPROPERTY)
(T (CH.PROPERTY SECONDARYPROPERTY)))
(PARSE.NSNAME NAME)
(CH.GETAUTHENTICATOR)
' RETURNERRORS])
```

(CH.ADD.SELF

```
[LAMBDA (GROUPNAME PROPERTY STREAM) (* jwo%: "7-Jul-85 17:24")
(SETQ GROUPNAME (PARSE.NSNAME GROUPNAME))
(COURIER.CALL (OR STREAM (CH.FINDSERVER GROUPNAME))
' CLEARINGHOUSE
' ADD.SELF GROUPNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(CH.GETAUTHENTICATOR T)
' RETURNERRORS])
```

(CH.DELETE.SELF

```
[LAMBDA (GROUPNAME PROPERTY STREAM) (* jwo%: "7-Jul-85 17:24")
(SETQ GROUPNAME (PARSE.NSNAME GROUPNAME))
```

```
(COURIER.CALL (OR STREAM (CH.FINDSERVER GROUPNAME))
' CLEARINGHOUSE
' DELETE.SELF GROUPNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(CH.GETAUTHENTICATOR T)
' RETURNERRORS])
```

)

(DEFINEQ

(CH.RETRIEVE.ITEM

```
[LAMBDA (OBJECTPATTERN PROPERTY INTERPRETATION) (* bvm%: "28-Jun-84 16:37")
```

```
:: Retrieves item property PROPERTY of object OBJECTPATTERN, returning (CanonicalName Value); If INTERPRETATION is given, then Value is
:: interpreted via it, otherwise Value is just a SEQUENCE of UNSPECIFIED
```

```
(SETQ OBJECTPATTERN (PARSE.NSNAME OBJECTPATTERN))
(PROG ((ADDRESS (CH.FINDSERVER OBJECTPATTERN T))
NAME&VALUE)
[COND
((AND ADDRESS (SETQ NAME&VALUE (COURIER.EXPEDITED.CALL ADDRESS \CH.BROADCAST.SOCKET ' CLEARINGHOUSE
'RETRIEVE.ITEM OBJECTPATTERN (CH.PROPERTY PROPERTY)
(CH.GETAUTHENTICATOR)
' NOERROR))
INTERPRETATION)
(RPLACA (CDR NAME&VALUE)
(COURIER.READ.REP (CADR NAME&VALUE)
' CLEARINGHOUSE INTERPRETATION)]
(RETURN NAME&VALUE]))
```

(CH.ADD.ITEM.PROPERTY

```
[LAMBDA (OBJECTNAME PROPERTY VALUE INTERPRETATION STREAM) (* Imm " 9-Jan-86 02:12")
```

```
(SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME))
(COURIER.CALL (OR STREAM (CH.FINDSERVER OBJECTNAME))
' CLEARINGHOUSE
' ADD.ITEM.PROPERTY OBJECTNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(COND
(INTERPRETATION (COURIER.WRITE.REP VALUE ' CLEARINGHOUSE INTERPRETATION))
(T VALUE))
(CH.GETAUTHENTICATOR T)
' RETURNERRORS])
```

(CH.CHANGE.ITEM

```
[LAMBDA (OBJECTNAME PROPERTY NEWVALUE INTERPRETATION) (* jwo%: "10-Jun-85 21:07")
```

```
(COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME)))
\CH.BROADCAST.SOCKET
' CLEARINGHOUSE
' CHANGE.ITEM OBJECTNAME (OR (FIXP PROPERTY)
(CH.PROPERTY PROPERTY))
(COND
(INTERPRETATION (COURIER.WRITE.REP NEWVALUE NIL INTERPRETATION))
(T NEWVALUE))
(CH.GETAUTHENTICATOR T)
' RETURNERRORS])
```

)

(DEFINEQ

(CH.LIST.DOMAINS

```
[LAMBDA (DOMAINPATTERN) ; Edited 24-Jul-87 17:53 by bvm:
```

```
(SETQ DOMAINPATTERN (PARSE.NSNAME DOMAINPATTERN 2))
(COURIER.CALL [CAR (CAR (fetch OCALLSERVERS of (\CH.FIND.ORG.SERVER (fetch NSORGANIZATION of DOMAINPATTERN)
T)]
' CLEARINGHOUSE
' LIST.DOMAINS DOMAINPATTERN ' (CLEARINGHOUSE . DOMAIN)
(CH.GETAUTHENTICATOR)
' RETURNERRORS])
```

)

```
:: NOTE: the following functions were moved to the file AUTHENTICATION temporarily for benefit of Lyric users wanting a functional NSMAINTAIN:
:: CH.LIST.PROPERTIES CH.LIST.ORGANIZATIONS CH.LIST.OBJECTS CH.DELETE.MEMBER CH.ADD.MEMBER CH.ADD.GROUP.PROPERTY
```

:: Other entries

(DEFINEQ

(LOOKUP.NS.SERVER

```
[LAMBDA (NAME TYPE FULLFLG) (* bvm%: "27-Aug-86 10:21")
```

```
:: Return the NS address of the specified server. If a type is given, use the Clearinghouse if the address isn't in the cache.
```

```
(SETQ NAME (PARSE.NSNAME NAME))
```

```
(PROG (ADDRESSES NAME&VALUE ENTRY)
(COND
  ((find old ENTRY in NS.SERVER.NAMES.TO.ADDRESSES suchthat (EQUAL.CH.NAMES (CAR ENTRY)
                                                                NAME)))
  [[SETQ NAME&VALUE (COND
    (TYPE (CH.RETRIEVE.ITEM NAME TYPE 'NETWORK.ADDRESS))
    (T (CH.RETRIEVE.ITEM NAME (CH.PROPERTY 'ADDRESS.LIST)
      'NETWORK.ADDRESS.LIST)
      [push NS.SERVER.NAMES.TO.ADDRESSES (SETQ ENTRY (CONS (CAR NAME&VALUE)
                                                           (SETQ ADDRESSES (COND
                                                               (TYPE
                                                                ; Old way
                                                                (LIST (CADR
                                                                NAME&VALUE
                                                                )))
                                                                (T (CADR NAME&VALUE]
                                                                (COND
                                                                (*ASSUME-ZERO-NSSOCKETS*
                                                                ; Stupidity. Old clearinghouses sometimes give a nonzero
                                                                ; socket here. Why?
                                                                (for ADDR in (CDR ENTRY) do (replace NSSOCKET of ADDR with 0]
                                                                (T (RETURN)))
                                                                (RETURN (COND
                                                                (FULLFLG ENTRY)
                                                                (T (CADR ENTRY]))
                                                                (COND
                                                                (*ASSUME-ZERO-NSSOCKETS*
                                                                ; Stupidity. Old clearinghouses sometimes give a nonzero
                                                                ; socket here. Why?
                                                                (for ADDR in (CDR ENTRY) do (replace NSSOCKET of ADDR with 0]
                                                                (T (RETURN)))
                                                                (RETURN (COND
                                                                (FULLFLG ENTRY)
                                                                (T (CADR ENTRY]))
```

(CANONICAL.NSHOSTNAME

```
[LAMBDA (HOST) (* bvm%: " 5-Feb-85 18:37")
  (PROG [(INFO (OR (LOOKUP.NS.SERVER HOST NIL T)
                  (LOOKUP.NS.SERVER HOST 'FILESERVER T]
  (RETURN (AND INFO (MKATOM (NSNAME.TO.STRING (CAR INFO)
T]))
```

(CH.CANONICAL.NAME

```
[LAMBDA (NAME) (* bvm%: " 4-Jul-84 14:46")
```

::: Canonicalizes name, unless it is a pattern. Result is a standard 3-part name. Name is not required to be a valid Clearinghouse name, but if it is, it is
::: dereferenced to its distinguished name

```
(PROG ((PARSED (PARSE.NSNAME NAME)))
  (RETURN (OR (AND (NOT (STRPOS '*' NAME))
                  (CH.LOOKUP.OBJECT PARSED))
              PARSED))
```

)

(RPAQ? *ASSUME-ZERO-NSSOCKETS* T)

(FILESLOAD AUTHENTICATION)

(PUTPROPS CLEARINGHOUSE COPYRIGHT ("Venue & Xerox Corporation" 1984 1985 1986 1987 1988 1990 1993))

FUNCTION INDEX

CANONICAL.CH.NAME	10	CH.ISMEMBER	12	GETCLEARINGHOUSE	6
CH.ADD.ITEM.PROPERTY	13	CH.LIST.ALIASES	11	LOOKUP.NS.SERVER	13
CH.ADD.SELF	12	CH.LIST.ALIASES.OF	12	NSNAME.TO.STRING	5
CH.CANONICAL.NAME	14	CH.LIST.DOMAINS	13	PARSE.NSNAME	9
CH.CHANGE.ITEM	13	CH.LOOKUP.OBJECT	12	SHOW.CLEARINGHOUSE	7
CH.CREATE.ALIAS	11	CH.NAME.TO.STRING	10	START.CLEARINGHOUSE	6
CH.CREATE.OBJECT	11	CH.PROPERTY	10	\CANONICAL.NSHOSTNAME	14
CH.DELETE.ALIAS	11	CH.RETRIEVE.ITEM	13	\CH.CHECK.CLEARINGHOUSE	6
CH.DELETE.OBJECT	11	CH.RETRIEVE.MEMBERS	12	\CH.FIND.ORG.SERVER	7
CH.DELETE.PROPERTY	12	CH.SERVERS	10	\CH.GUESS.NEW.PROPERTIES	11
CH.DELETE.SELF	12	COURIER.NSNAME.LENGTH	6	\CH.LOCATE.SERVERS	8
CH.DOMAINS.SERVED	11	COURIER.READ.NSNAME	5	\CH.PRINT.BLURB	8
CH.FINDSERVER	7	COURIER.WRITE.NSNAME	6	\CH.UPDATE.CACHE	8
CH.GETAUTHENTICATOR	10	EQUAL.CH.NAMES	9	\NSNAME.DEFPRINT	5

VARIABLE INDEX

ASSUME-ZERO-NSSOCKETS	14	CH.PROPERTIES	4	NS.SERVER.NAMES.TO.ADDRESSES	4
CH.DEFAULT.DOMAIN	4	CLEARINGHOUSE.STRUCTURE.WINDOW	4	\CH.CACHE	4
CH.DEFAULT.ORGANIZATION	4	CONSTANTFOLDERS	4	\CH.MAINLOCK	4
CH.NET.HINT	4	LOCAL.CLEARINGHOUSE	4	\SYSTEMCACHEVARS	4

RECORD INDEX

CHBROADCAST	4	DOMAINCACHE	4	NSNAME	5	ORGCACHE	4
-------------------	---	-------------------	---	--------------	---	----------------	---

PROPERTY INDEX

NSNAME	5	NSNAME2	5
--------------	---	---------------	---

CONSTANT INDEX

\CH.BROADCAST.SOCKET	4	\CH.NULL.PROPERTY	4
----------------------------	---	-------------------------	---

OPTIMIZER INDEX

CH.PROPERTY	4
-------------------	---

TEMPLATE INDEX

CH.PROPERTY	4
-------------------	---

COURIERPROGRAM INDEX

CLEARINGHOUSE	2
---------------------	---
